DOCUMENT RESUME

ED 057 808 LI 003 297

AUTHOR	Briggs, Bruce
TITLE	Center for Information Services, Phase II: Detailed System Design and Programming, Part 3 - Software System Design and Related Software Activities, Phase IIA Final Report.
INSTITUTION	California Univ., Los Angeles. Inst. of Library Research.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	1 Apr 71
NOTE	63p.; (0 References)
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Programs; *Design; *Information Centers; *Information Processing; Information Retrieval; *Information Services
IDENTIFIERS	*Computer Software; University of California (Los Angeles)

ABSTRACT

During Phase IIA a component of the Center for Information Services (CIS) project was established within the Campus Computing Network to develop the software needed by CIS. The primary task of this component was to develop a basic design of the Information Processing System for CIS. This report describes the status of the design efforts at the end of Phase IIA. Other tasks involved the development and installation of software related to, or in support of, the principal software system. Reports on these tasks are included also. (For related documents see LI 003295, 003296 and LI 003298 through LI 003301). (Author)

ED057808

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

CENTER FOR INFORMATION SERVICES

PHASE II: DETAILED SYSTEM DESIGN AND PROGRAMMING

NSF GRANT GN-827

PHASE IIA FINAL REPORT

PART 3

SOFTWARE SYSTEM DESIGN AND RELATED SOFTWARE ACTIVITIES

by

Bruce Briggs

1 April 1971

Institute of Library Research
University of California
Los Angeles, California

LI 003 297

63p.

TABLE OF CONTENTS

		PAGE
	LIST OF ILLUSTRATIONS.	iv
	ABSTRACT	v
	ACKNOWLEDGMENT	vi
I.	INTRODUCTION	1
	Background.	1
	Design Decisions.	3
	Definitions	5
II.	CIS INFORMATION PROCESSING SYSTEM.	9
	General Description	9
	Program Structures.	11
III.	RELATED SOFTWARE DEVELOPMENT ACTIVITIES	20
	Interim Software.	20
	User Interface.	20
	FMS	22
	DISCUS.	23
	Numerical Processing Studies.	23
IV.	EXPERIMENTAL SDI SERVICE	24
	Software.	24
	Operating Procedures.	24
	Promotion	24
	User Interface.	25
	Monitoring and Evaluation	25
	Administration.	26
	Implementation Schedule	26

TABLE OF CONTENTS (Continued)

	PAGE
APPENDIX A - CHEMICAL ABSTRACTS OPERATING PROCEDURES. . .	27
APPENDIX B - ANNOUNCEMENT OF EXPERIMENTAL SERVICE	36
APPENDIX C - USER STATEMENT OF INTEREST	38
APPENDIX D - WRITING TEXT-PAC PROFILES.	42
APPENDIX E - INITIAL USER INTERVIEW QUESTIONNAIRE	48
APPENDIX F - FOLLOW-UP QUESTIONNAIRES	51
APPENDIX G - USER PROFILE RECORDS	54
APPENDIX H - CA-CONDENSATES SEARCH SUMMARY.	56

LIST OF ILLUSTRATIONS

FIGURE		PAGE
1	STRUCTURE OF A TYPICAL SIMPLE SERVICE.	13
2	TYPICAL PACKET-STRUCTURE FRAGMENT CORRESPONDING TO A SIMPLE SERVICE.	14
3	A TYPICAL JOB MONITOR.	16
4	A TYPICAL COMPLEX SERVICE--KWIC.	17
5	TASK STRUCTURE	18
6	APPROXIMATE PACKET STRUCTURE	19
7	OPERATIONAL CIS - HANDLING USER REQUESTS	21
G-1	USER PROFILE RECORDS	54
H-1	CA CONDENSATES SEARCH SUMMARY.	56

ABSTRACT

During Phase IIA a component of the CIS project was established within the Campus Computing Network to develop the software needed by CIS. The primary task of this component was to develop a basic design of the Information Processing System for CIS. This report describes the status of the design efforts at the end of Phase IIA. Other tasks involved the development and installation of software related to, or in support of, the principal software system. Reports on these tasks are included also.

ACKNOWLEDGMENT

The following members of the CCN staff are assigned wholely to the CIS project and participated in the Phase IIA efforts described herein.

CIS Software Development Group

Barbara Blankenship*
Aeint H. de Boer
R. Bruce Briggs
Pete Donahoe
William J. Jordan
Noelleen G. Kolb
Harrell C. Ludlam
Linda J. Miroff
Jerry M. Pine
Ida B. Riordan
Steven S. Silver*

Four programmers (de Boer, Miroff, Riordan, and Silver) participated in CIS as members of the Institute of Library Research prior to joining the CCN group.

*No longer with the project.

I. INTRODUCTION

During the first half of Phase IIA of the Center for Information Services (CIS) development project, members of the Institute of Library Research (ILR) experimented with a number of data bases to derive information that could be translated into requirements, specifications, software system designs, operating procedures, and evaluation approaches. Early in 1970 a component of the CIS project was established within the Campus Computing Network (CCN) to develop the software needed by CIS. The CCN group began developing the basic design of the software, drawing upon the information and experience gained earlier by ILR. In mid-1970 the programmers at ILR were transferred to CCN to consolidate the design effort.

The primary task of this CIS Software Development Group (see Acknowledgment) has been to develop the basic design of the Information Processing System (IPS) for CIS as described in Section II. The group has been involved in a number of related activities which includes producing a text-formatting system (FMS), work on an interactive instructional package (DISCUS), and installation of interim software systems. The purpose of the interim software is to provide limited information processing capabilities until the IPS is operational. The interim software includes a set of programs developed by Aeint de Boer as part of his Master's degree requirements and an IBM package called TEXT-PAC. Mr. de Boer's programs are being used to process the ERIC data base and are described in Part 1 of the Phase IIA Final Report. TEXT-PAC will be used to provide an experimental SDI service (see Section IV) to the Chemistry Department and will use the CA-Condensates file supplied by the Chemical Abstracts Service.

BACKGROUND

CCN's primary role during Phase IIA was to produce the basic design of the software system, known as the Information Processing System (IPS), for CIS. As a first step in the design a set of general requirements were specified.

1. The IPS must consist of a variety of services capable of processing bibliographic, textual, and numerical data.

2. The IPS must have a built-in flexibility to add new services and change existing ones.

3. It must be data base independent; i.e., it must be able to handle the many existing formats as well as currently unforeseen ones.

4. The IPS should be able to service users in both a batch and an interactive mode.

5. It should have a built-in instructional capability.

6. The IPS must have built-in accounting and monitoring functions.

7. It must be efficient and cost-effective.

8. It should be exportable.

With these general requirements in mind, the basic design of the IPS evolved through a study and decision making process based on experimentation.

The first computer-related efforts of the CIS project consisted mainly of experimentation (reading, reformatting, and printing) with a variety of sample tape data bases. This was done to get some first-hand experience with the different kinds of data bases. It was soon discovered that the only characteristic common to all these data bases was that they were all sequential files. There appeared to be as many different record structures as there were distributors of tape files.

This lack of standardization presented the first problems to be solved in deriving a comprehensive information retrieval system. Should the system philosophy be that all data bases are first converted to a common format, or should the system be general enough to read any kind of file? If the former approach is taken should the converted file be sequential or inverted? If the latter approach is taken, what method should be used to make the system general? The former was rejected since much of the work done by CIS will be of the current awareness nature, which means that, in general, each file will be processed once and then not used, except for an occasional retrospective search. Hence, it appears that the cost of converting a file to a common format is not warranted.

There are several ways in which the second approach can be implemented; these include such methods as a compiler that can generate programs to handle each new data base, or a general file description language in which any given file can be described. The approach CIS has decided to implement is a simple one: a read routine will be written for each data base structure the system needs to use. The idea behind this approach is that the bulk of the system can be standard, and a read routine acts as an interface between a given data base and the

rest of the system. It was decided that development of a general-purpose program to read all structures would divert too much effort from the primary task: namely, development of the IPS. The need for a general-purpose program will be reconsidered at some later time.

The next problem that CIS examined was that of the choice of a programming language. A study was made of the several languages available under OS/360 and it was decided that PL/I was the best choice, at least as a language for development. For reasons of efficiency it may be necessary to rewrite portions, or the entire system, in Assembler Language for the final version of the system.

Having made the above decisions, an overall scheme for the information-processing system was derived. This design was described in the Fourth Quarterly Progress Report for Phase IIA. It consisted of a Monitor, an Input Module, an Output Module, a Translator, an Analyzer, and a Process Manager. Experimental versions of the components were constructed and evaluated, and as a result, several problems were discovered, most of which were related to insufficient specification of detail. This necessitated a re-examination of the system requirements.

To accomplish this examination, the staff was divided into two teams. The first re-examined the service requirements, and the second re-examined the technical considerations of the IPS. This resulted in a revised version of the IPS, consisting not only of a slight restructuring of the overall model, but allowing detailed specifications to be established more clearly. The revised version, described in Section II, is based on the design decisions and definitions which follow.

DESIGN DECISIONS

A number of design issues were considered and the following decisions were made to guide the IPS design effort.

1. Host System: The Information Processing System will be written to operate under the MVT level of the OS/360 executive. Except for application-oriented subprograms, it will require only the System/360 standard instruction set, and will assume its host machine to demand rigid operand alignment.

2. Datasets: Datasets used by the IPS will be completely compatible with most system formats, and can be used with or without the assistance of the IPS. For this reason, dataset security must be provided by the host system, and cannot be a part of the IPS. The IPS will, however, be subject to the host system's dataset security system.

3. Subsystem Residence and Access: The IPS will be useable in both an interactive and a batch mode; however, in either case each

user will be a 'job' to the operating system, so the IPS will not require an operator-managed application-dependent resident monitor.

4. Supervisor Embellishments: Operation of the IPS will not require any additions or changes to the executive system. No IPS-provided code may operate in supervisor state, or memory-protect-privileged.

5. Exportability: Every effort will be made to make the IPS quickly implementable on any machine running the required executive system. Such implementations will not necessarily be as efficient as the UCLA implementation.

6. User Programming Interface: Processing programs can be written by the user and executed under the IPS, provided certain conventions are followed. These conventions will be documented for both Assembler and PL/I programmers. Because of the complexity of the environments required by the running systems attached to most high-level language translators, code produced from such languages cannot be integrated into a controlled environment such as required by the CIS IPS except by close cooperation on the part of the IPS. This cooperation will be explicitly provided only for the IBM PL/I-F programming system.

7. Accounting: The IPS will maintain a subsystem statistical reporting log, from which user-billing information can be extracted. However, this facility will not be used for the billing of the subsystem itself by the host computing facility. This function will remain in the hands of the systems personnel for the host system. The two statistical systems will not interact.

8. User Billing: The vehicle for billing subsystem users will be the IPS' subsystem statistical log. Billing will be based on factors directly under the control of the user, and not on those factors used by the host installation to bill the subsystem for computer utilization. The latter set is specifically inappropriate for the former purpose, due to the IPS concept of File-sharing (see below).

9. Time-sharing: In order to avoid incompatibilities with any of the various time-slicing and roll-out schemes in use in various systems, the IPS will not provide its own version of either of these services. It will be the responsibility of the host system to effect commutation, resource saturation control, and core-sharing, to the extent and by the method indicated by the environment in which the IPS will be run. In the UCLA implementation, these functions will be filled, in so far as possible, by the IBM Time Sharing Option (TSO) of OS/360.

10. Modularity: All programs written for the IPS will be designed modularly, such that changes in individual environmental parameters will affect a minimum number of modules. Also, modules for the same basic function will be shared among system applications wherever possible.

11. Dynamic Generation: The binding of program modules into executable programs will be done individually for each execution, and will be governed by analysis of the environment at execution time.

12. File Independence: All routines handling data files will do so only through the use of IPS-provided subroutines. The interface to these routines is designed to make the processing routines independent of the details of file format. Actual format-dependent code will be selected, according to the file being processed, through the dynamic generation facilities.

13. File Sharing: The basic system design of the IPS will include a facility for the concurrent execution of multiple independent processing requests against one sequential data file, during a single pass of that file. This facility will be transparent to the processing programs of the IPS, and transparent to the user. Because of this facility, computer resource utilization statistics will be inappropriate factors in billing users of the subsystem, and a scheme based on user-controllable factors must be developed.

DEFINITIONS

The following definitions were developed for CIS use:

Load Module and Packet

Load Module: The basic unit of executable code, and the smallest such unit manipulated by the CIS Monitor. Load modules are produced by the OS Linkage Editor.

Packet: A collection of load modules which are associated together--either because they use each other during execution or because they are all used by a single caller. A packet has a usage characteristic (either open or closed) and a generation characteristic (either dynamic or linked).

Closed Packet: A packet representing a group of interdependent load modules, of which one is designated 'main'. Execution of a closed packet may only he initiated by calling the main member. All other packet members must execute subordinate (not necessarily immediately) to the main member. A closed packet is thus a mechanism for execution-time binding of inter-modular dependencies.

Open Packet: A packet representing a group of independent load modules, each of which may be executed directly by code outside the packet. An open packet thus represents a simple vector of procedure, and is useful in any situation where it is convenient to refer to algorithms by number.

Packet Control Block (PCB) and Slot

Packet Control Block (PCB): The internal representation of a packet, consisting basically of a vector of load-module names and entry points. The PCB provides both the defining mechanism for a packet, and the mechanism by which members are referenced.

Slot: An element of the PCB, considered as a vector. Each slot of a particular packet normally corresponds to a specific function, but may be occupied by any one of a set of modules which are capable of performing that function.

Named Slot: A slot of a PCB into which has been placed the character name of a CIS load module which has been selected to occupy the slot. Named slots are the only ones which are processed by the CIS Monitor's packet-management services.

Blank Slot: A slot of a PCB whose name field has been set to a character blank value. Blank slots are not processed by the CIS Monitor's packet-management services, and their maintenance is the responsibility of the routines generating the packet.

Current PCB: The PCB which is being used by the routine executing at the moment for accessing CIS Monitor services. The notion of a current PCB is necessary in order to maintain some order when the number of active PCB's becomes large.

Ascendent PCB: The PCB which was current when the PCB in question was generated, and to which its link field thus points. Ascendence is a relation betwee.. two PCB's.

Descendent PCB: Any PCB which was generated when the PCB in question was current, and which is thus linked to it. Descendence is a relation between two PCB's.

PCB Tree: The structure of PCB's which is defined by the notions of ascendence and descendence. This structure naturally assumes the shape of a simply-branching tree rooted in the main PCB for the Job Monitor.

Packet Generation Characteristics

Packet Generation: The process (supplied as a CIS Monitor Service) during which load modules are actually collected, brought into core, and bound into a packet as members.

Packet Generator: A routine which assumes the responsibility for selecting the modules which will occupy the slots of a new packet, filling in name fields appropriately, and calling the CIS Monitor's packet-management routines to generate, and later to erase, the new packet. Any routine whatsoever may assume such responsibility, and thus become a Packet Generator.

Dynamic Generation: That method of Packet Generation which only associates load modules. The load modules are not modified in any way, do not lose their external identification, and thus do not become dependent on the packet. In this way, a single copy of such a module may be a member of any number of dynamically-generated packets simultaneously. A dynamically-generated packet may be either open or closed, but a closed dynamically-generated packet using PL/I-generated subordinate members may have a prohibitively high CPU overhead during execution.

Linked Generation: That method of packet generation which binds together new copies of load modules into a single procedural entity associated with one PCB. This is basically the same as using the Linkage Editor to produce a single new load module composed of the union of the packet member load modules, meaning that inter-module linkages may be established. Since this implies that the modules may be altered during the process of generation, their individual identities are not known outside the generated packet, and thus they may not be shared among packets except by using separate copies. Linked packets are usually closed, especially if any PL/I-generated load modules are included.

Module Lending: The technique, used by packet generators, of filling a slot by making it a blank slot and copying an entry point from a previously-generated PCB. The lending packet must be an ascendent (not necessarily immediate) of the borrower. Module Lending is usually used to propogate the entry addresses of the Job Monitor and the User Communicator routines through all PCB's. It may be used for any slots, however.

Subsupervision: A technique, available to packet generators, by which activity specific to a slot of the new packet may be monitored and/or modified by code on the generator's level. This is normally accomplished by providing the usual routine for a slot through a higher-level PCB, and filling the corresponding slot of the new PCB with a monitoring routine, which can call the original at its discretion. Subsupervision is normally used to effect input file sharing, and to provide internal replies from the User Communicator slot.

Services

Program Services: A set of service routines which are called by component programs of the IPS to effect processing best centralized

for the job. IPS program services are thus of interest only to the programmer, and not the IPS user. The program services are categorized as Calling Services, Communicator Services, and Job Monitor Services.

User Service: A well-defined set of major operations to be performed for the user, under any of a variety of circumstances. When not otherwise qualified, 'service' usually means 'user service' as opposed to any of the types of program services. Services are named, and are usually associated with similarly-named major commands which may be issued by the user.

Subservice: A service which is being executed subordinate to another service, called the superservice. Theoretically, any service can be executed as a subservice, without any effects on its operation.

Superservice: A service which executes other services as subservices.

Complex Service: The structure formed by a superservice and the ordered set of subservices which it uses. A complex service may include several levels of subservices.

II. CIS INFORMATION PROCESSING SYSTEM

GENERAL DESCRIPTION

The Information Processing System of the Center for Information Services or (CIS IPS) appears to the casual user to be a single computer program, invoked through the standard system command languages. This program accepts input in a command language of its own, and performs requested operations on datasets which are libraried at the computer site. In fact, the IPS consists of a large collection of programs, libraries of metacommand text and dataset descriptors, and several catalogued procedures to make the IPS quickly accessible through a minimum of host system commands.

The IPS can be used through either of two computer access methods, batch-jobbing access or interactive access. When used through batch-jobbing access, the user usually specifies his processing requirements on a deck of punched cards, or some equivalent medium, and the computing system returns results in the form of printed listings and/or punched cards. When used through interactive access, the user converses with the computing system through some form of keyboard terminal, which may or may not provide hard-copy output.

Host-System Commands

For either of the two computer access methods, batch-jobbing or interactive, the user must identify himself to the host system before he can communicate directly with the IPS. In interactive access, this requires only the typing of a simple LOGON command. In batch access, a JOB statement and an EXEC statement are required. After the host system is satisfied as to the identity of the user, it turns control over to the IPS's Job Monitor program, which handles communication with the user from then on. Because a different version of the Job Monitor is automatically used for batch versus interactive access, most of the differences between the two modes of access are handled internally, and need not concern the user.

IPS User Services

Once the user is communicating directly with the IPS, he enters commands in the IPS command language. This language is designed to be modular, so that a user need only learn those parts required to do the type of work that interests him. This is done through the notion of IPS user services. The user service is the element of the IPS which is most apparent to the user, being the entity which he requests by name through the use of the major commands accepted by the Job Monitor. Once a service has been

selected, it takes control of the input interpretation process, and, provided it generates no incompatibilities with the rest of the system, it may extend the acceptable command set in any way that is appropriate for the application in which it specializes.

An executing service may invoke any other service as a subservice. A subservice amy be allowed direct communication with the user and/or the Job Monitor, or the invoking service may subsupervise either or both these interfaces. In this way, information which is requested of the user by a service may actually be supplied internally by an invoking service, if it already has the information, and this is completely transparent to the subservice. This system prevents the necessity of a service coding its own version of processing, which could actually be done by an existing service except for the requirement that the user provide redundant information.

File Identification

Besides specifying the type of processing to be done, the user must also identify the input and output files to be used. If a file is one which has been entered into the IPS dataset indices by a System Controller, the processing programs will automatically locate programs in the program libraries which are capable of processing the particular sort of file formats that are involved. If the file is not so indexed, but has a format which has been named and described in the IPS descriptor libraries, the user may state the file name and the format name, and receive the same automatic services. In other cases, if the file is not given to strange idiosyncrasies, the user may provide a file description imbedded within his input text. It is hoped that most common file formats can be put in libraries because the description process can not be simplified to the point where the average user should be expected to have to learn it.

Autobatch Facility

Interactive access to the computing system corresponds to execution in the foreground area of the system environment. Foreground execution involves a system overhead very much greater than that experienced by jobs executing in the background, as do jobs using batch access. Not only is foregound execution expensive to the user, but it loads the computing system much more quickly than does background execution, thus degrading system performance for all users. For such reasons, it is desirable to restrict the work done in the foreground to that actually requiring interactive access for successful execution or worthwhile results. It is particularly desirable to avoid long, time-consuming operations in the foreground. At the same time, the entry of input parameters requesting such operations should be done interactively, to avoid reruns due to simple syntactic errors, which are costly both in terms of the user's time and the computer's. To address these problems, the IPS provides an autobatch service. This service can be invoked by any other service at the point

where input parameters have been collected and validated and actual processing would normally begin if, for any reason, the service decides that the request should be executed in the background. The autobatch service will move the collected input parameters to an IPS batching queue, where it will pend until a batch analysis run is made. The analysis run will be initiated by the System Controller, who will analyze the collected requests for ones which should be executed under the IPS's file-sharing supervisor, which is further discussed below. The collected requests are then entered through batch access. Output from such batched runs can be printed as is normal for background execution, or can be viewed during a later interactive session with the user.

File-Sharing Facility

The reason that foreground-entered requests for background execution are entered into a batching queue, rather than directly into the operating system's input work queues, is to provide an opportunity for use of the IPS' file-sharing facilities. File-sharing is a technique which allows multiple services to process the same data file during a single sequential pass of the file. When intelligently planned, file-sharing can save significant amounts of I/O activity; however, if used unwisely, it can merely add to the CPU time requirements of the batch. For this reason, the batched runs are first processed by an analysis service, executed by the System Controller. Through cooperative decision-making, the system and controller will form the collected requests into batches, according to whatever criteria the controller believes to be effective under the circumstances.

Each such batch will be identified with a common file name, and will be submitted under system command language statements which will execute the IPS file-sharing service instead of any of those requested. The file-sharing service will then be responsible for attaching the original list of services, and for maintaining order among them. Each such batch will constitute one host system job, and each request which cannot be batched to advantage will be entered as an individual job. Since the host system accounting is done by job, requests which are batched will not be separately accounted on this level. Proper user billing of such requests must be made on the basis of statistical output from the IPS' own statistical reporting log, which must be designed to fill this function. This implies that neither actual CPU time nor actual I/O counts may be used as a component of user billing, although they may be used for billing of the subsystem by the installation. This approach is compatible with the transparent batching concept.

PROGRAM STRUCTURES

Execution of the programs selected to service a user request is described by three interlocking tree structures. The service structure is a tree representing the inter-service relationships within a complex service. The packet structure is a tree representing the generated program packets

that exist at given moments during execution. The task structure is a tree defining concurrent control paths existing at a given moment. These structures interact, but are not usually identical.

The structure of an IPS User Service may vary widely; however, a typical structure such as the one diagrammed in Figure 1 shows that a service usually involves several levels of control. On the top level is the service router module, which directs the flow of control among the other components. This module is the one which is named by the LINK, ATTACH, or XCTL by which the service is invoked. It executes under the task which is current on entry, which, unless entered via ATTACH, is the caller's task. In all cases, it exits at the end of service processing via a simple RETURN.

On the second level, the router may pass control to a translator or an analyzer routine. Both of these routines are concerned with collecting and interpreting input from the user, and in some cases they may be a single routine. The distinction is convenient, however, as the translator, which is concerned only with syntax, is usually a common routine used system-wide, while the analyzer, which is concerned with interpretation in context, is more often supplied specially for the service or for a class of services. The router also calls generator and eraser routines to build and/or destroy the packet structure which usually comprises the third level of control within a service. Transfers between the router and the second-level modules are normally effected via LINK and RETURN, and thus these modules operate under the same task as the router.

The third level of control within a service usually consists of a single generated packet, although more than one is permitted. The packet is usually entered by ATTACH and exited by RETURN. It thus executes under a task different from the router's. This is recommended to minimize the effects of abnormal termination, which is more likely in a generated program than in one which always runs with the same combination of modules. The PCB for the packet is chained as a descendant of the one which was current on entry to the router.

The above components are representative of the algorithmic aspect of a service. In addition, there are usually a variable number of data files associated with a service excution. Associated with each such file is a group of routines selected to convert the item formats used in the file to standard internal ones, and a group of control blocks, including an open PCB for tying these routines together. These routines exist on a fourth control level within the service. They are entered via CALL, and so execute under the same task as the packet with which they are associated. The file PCB's (known as Item Status Tables, or IST's) are chained as descendants of the PCB for that packet.

Thus it is seen that service boundaries are not usually the same as packet or task boundaries, although there is a definite relation among all three as illustrated in Figure 2. The symbolic structure in Figure 2a is used in following figures to represent a typical service.

FIGURE 1

STRUCTURE OF A TYPICAL SIMPLE SERVICE

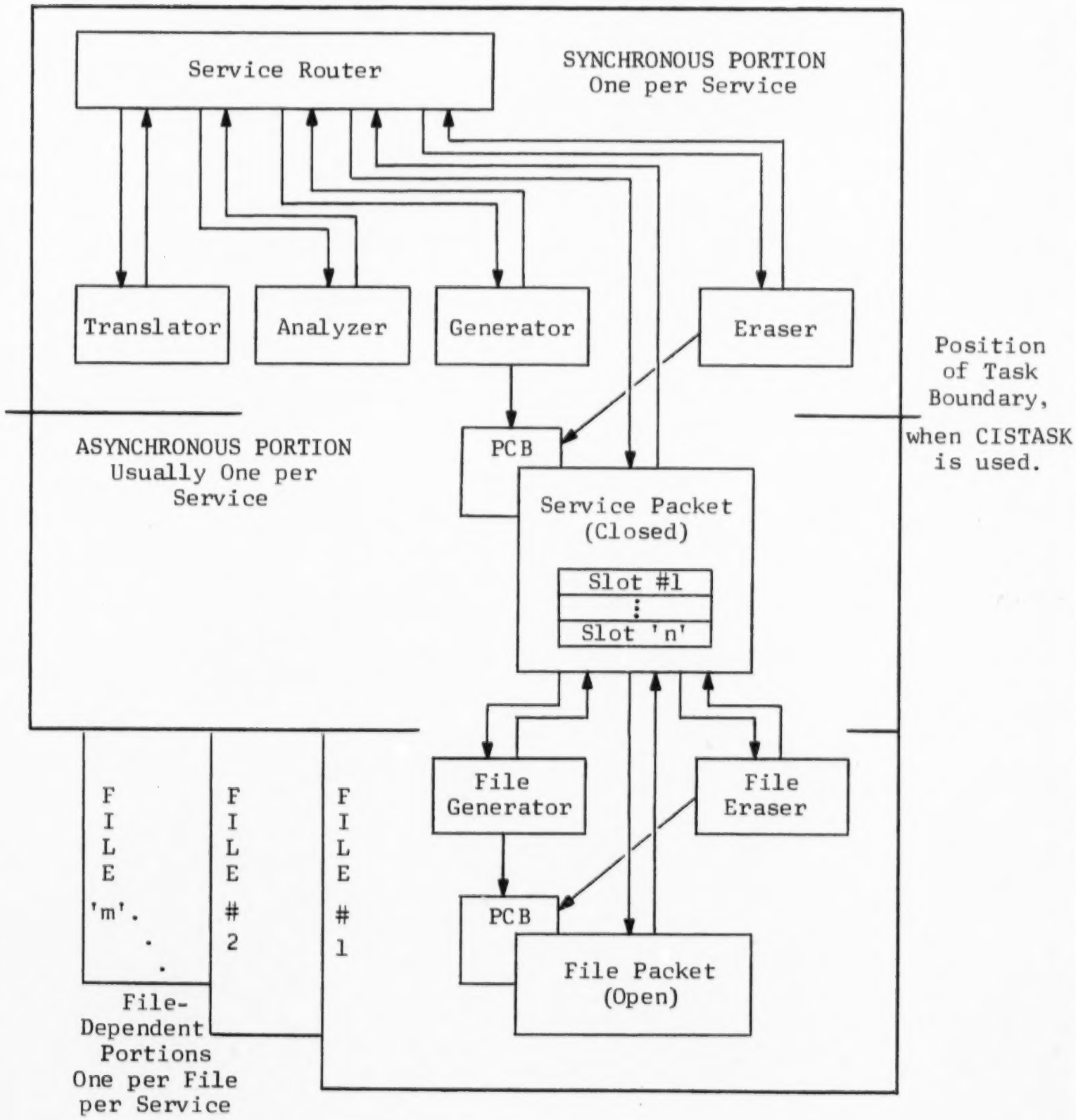
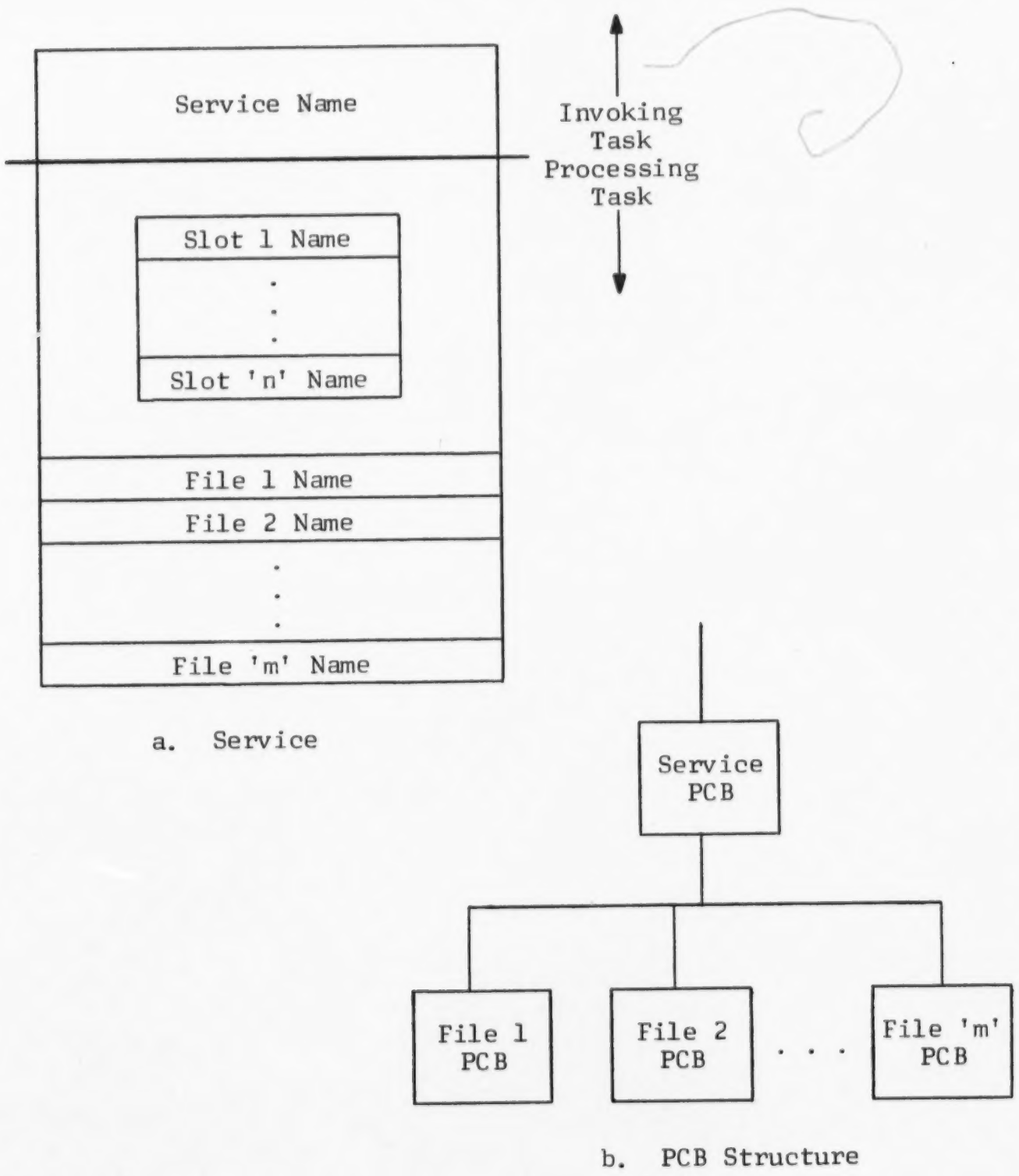


FIGURE 2

TYPICAL PACKET-STRUCTURE FRAGMENT CORRESPONDING TO A SIMPLE SERVICE



a. Service

b. PCB Structure

The roots of all three tree structures are found in the Job Monitor, which is the program executed by the host system when the IPS is first invoked. There are several versions of the Job Monitor, to take care of differences in computing system access methods, etc., but all appear basically as illustrated in Figure 3. The Job Monitor simulates a superservice from the aspects of subordinate services, and specifically builds the first PCB, which will be an ascendant of all PCB's built by executing services.

As an example of the structures generated by the execution of a complex service, Figure 4 illustrates the service structure of a hypothetical KWIC service. Figures 5 and 6 show the corresponding task and packet structures.

FIGURE 3

A TYPICAL JOB MONITOR

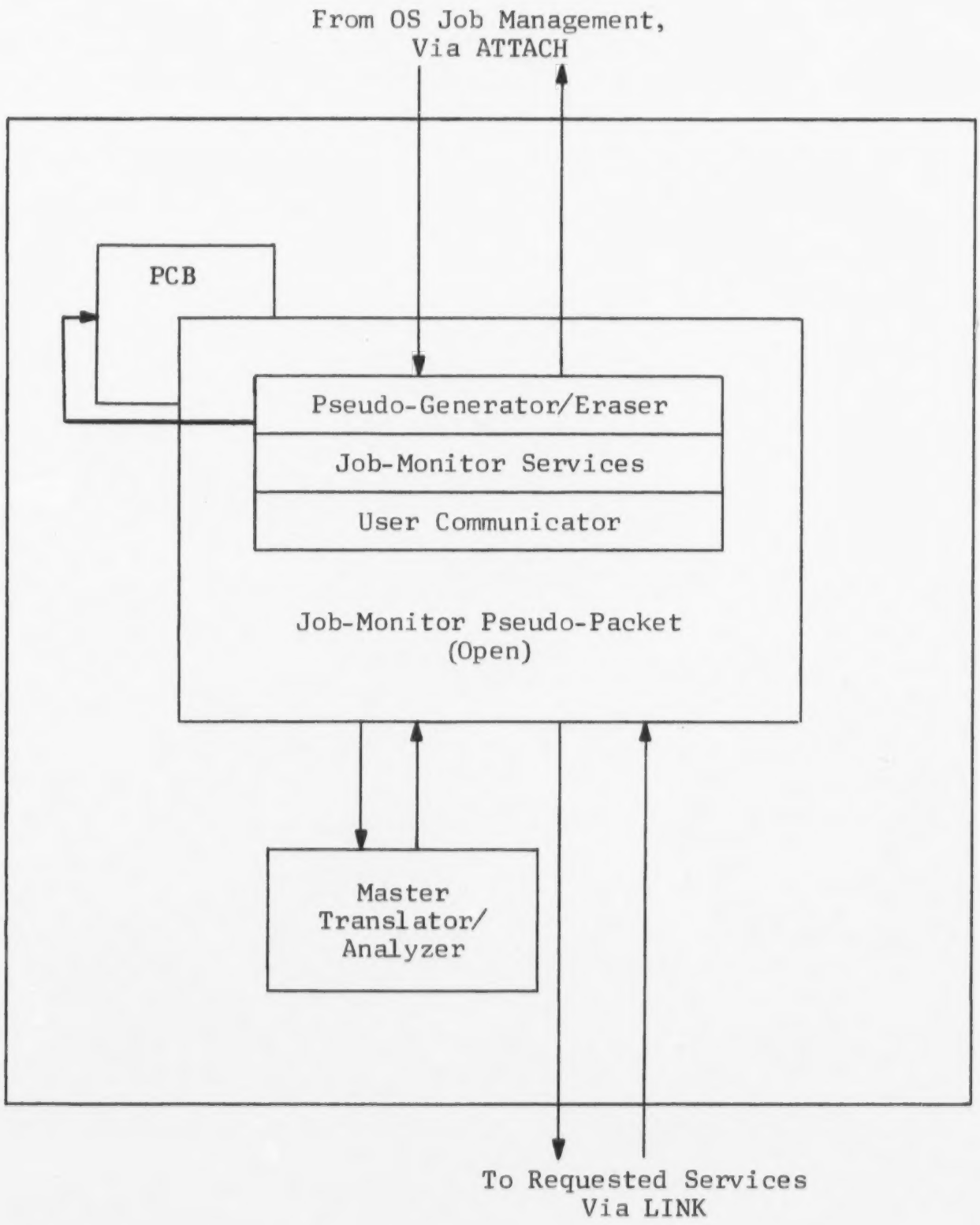


FIGURE 4

A TYPICAL COMPLEX SERVICE--KWIC

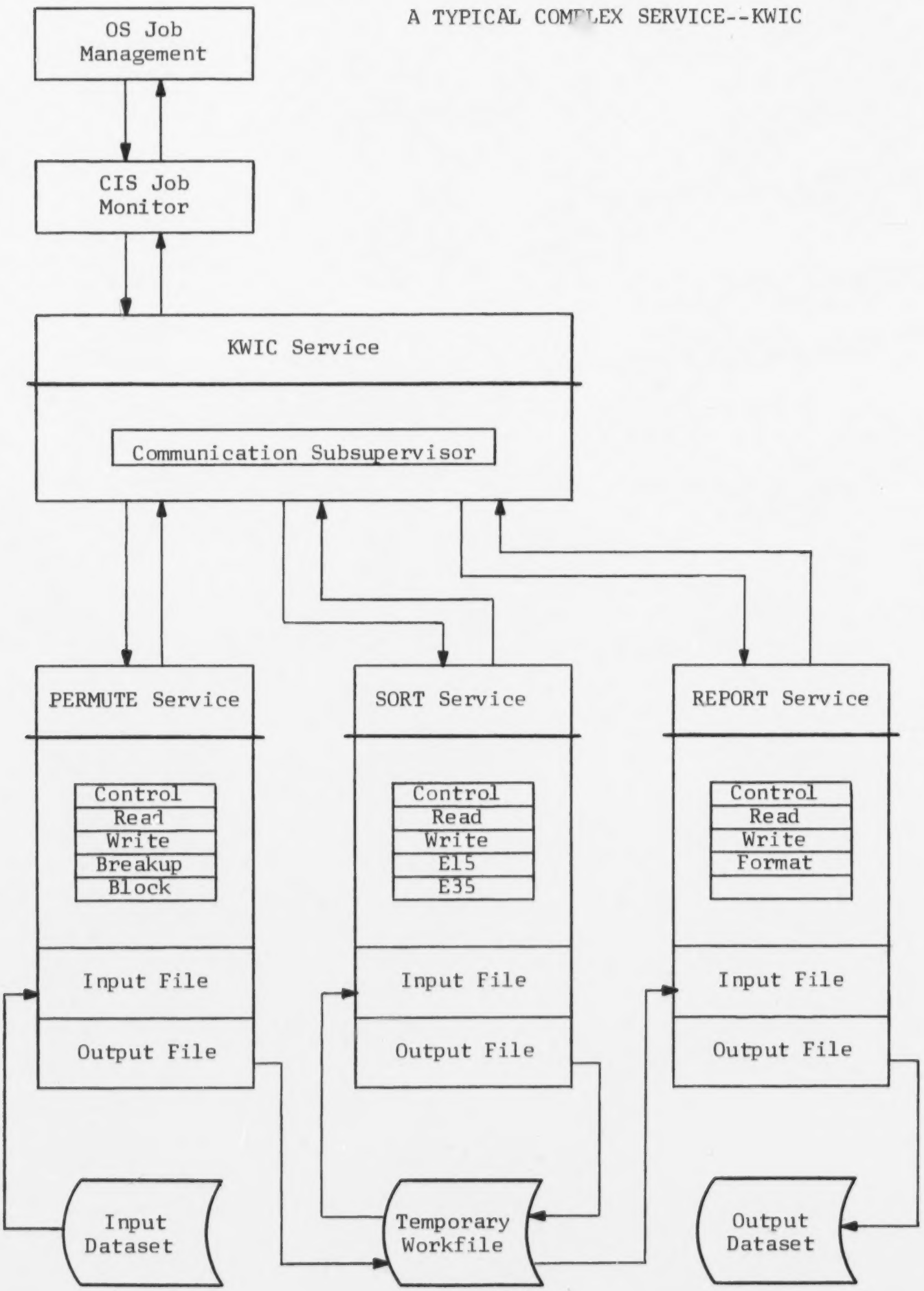
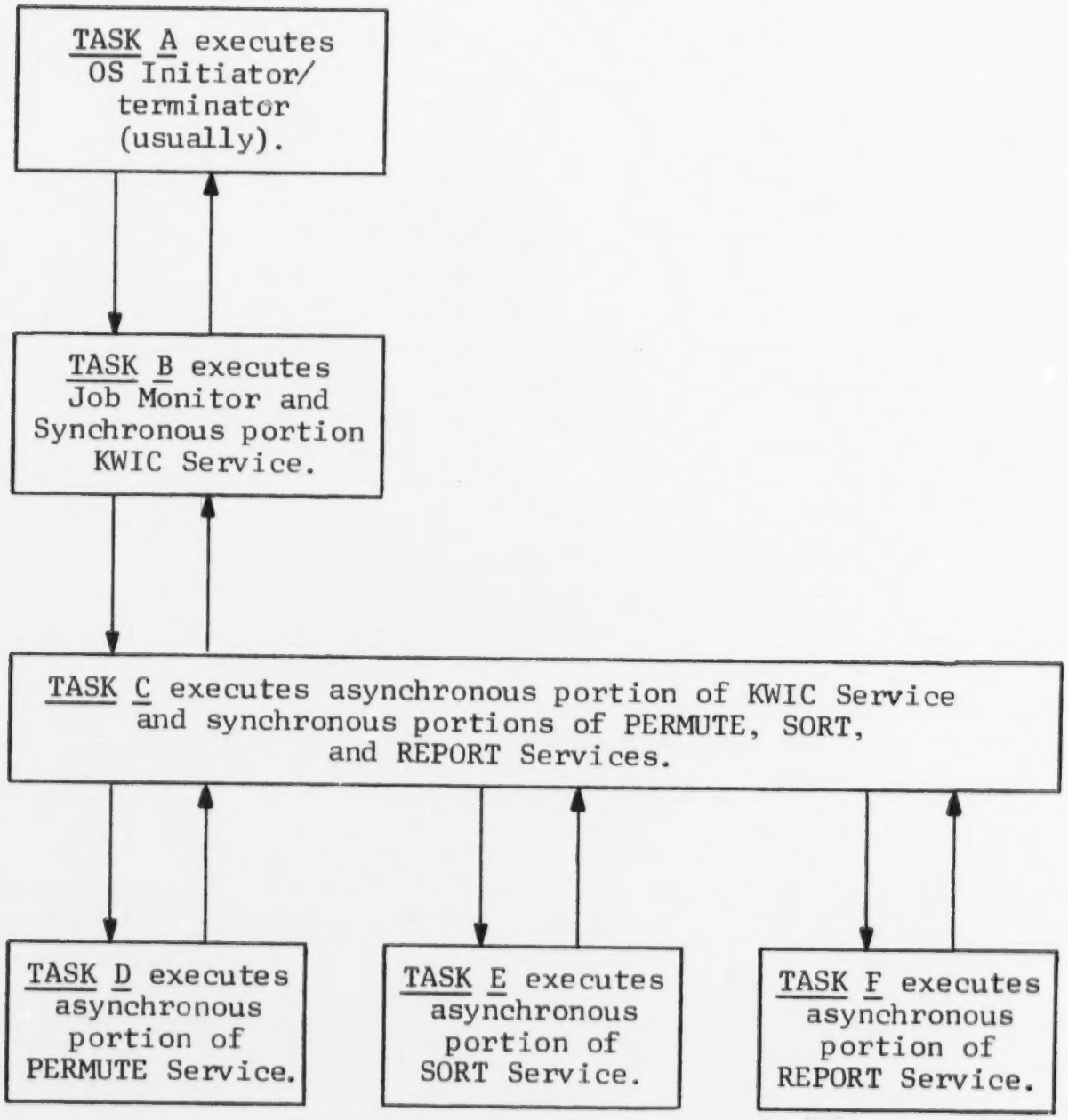


FIGURE 5

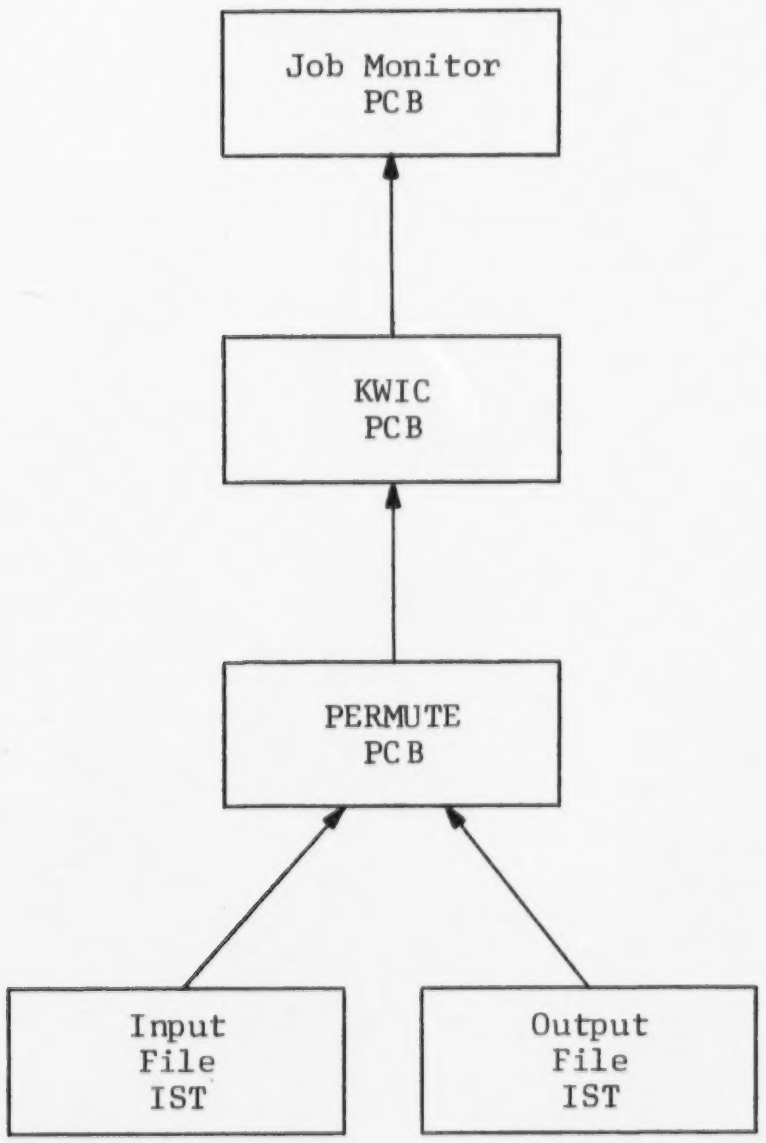
TASK STRUCTURE*


TASK A executes OS Initiator/ terminator (usually).
TASK B executes Job Monitor and Synchronous portion KWIC Service.
TASK C executes asynchronous portion of KWIC Service and synchronous portions of PERMUTE, SORT, and REPORT Services.
TASK D executes asynchronous portion of PERMUTE Service.
TASK E executes asynchronous portion of SORT Service.
TASK F executes asynchronous portion of REPORT Service.

* Corresponds to the KWIC Service Structure of Figure 4.

FIGURE 6

APPROXIMATE PACKET STRUCTURE*



* During PERMUTE phase of KWIC Service shown in Figure 4.

III. RELATED SOFTWARE DEVELOPMENT ACTIVITIES

INTERIM SOFTWARE

CIS will use existing software until the prototype of the IPS is developed. This software, referred to as interim software, may be modified to fit the needs of CIS, but major software development efforts not directly related to the IPS are not planned. The interim software being used presently by CIS consists of two packages: a set of information retrieval programs developed by Aeint de Boer, and a retrieval system, TEXT-PAC, developed by IBM.

Students in Professor Borko's class in the Graduate School of Library Service will use Mr. de Boer's programs to make searches of the ERIC file. The ERIC Clearinghouse for Junior Colleges and the Southwestern Regional Laboratory will also use the de Boer programs to offer SDI services from the ERIC file. Feedback will be obtained from all three groups on such factors as ease of use, search efficiency, flexibility, and reliability of the programs.

TEXT-PAC will be used by CIS to provide an experimental SDI service using the CA - Condensates file. Plans for this service are described in Section IV of this report. A program to convert CAS's Standard Distribution Format (SDF) to the TEXT-PAC format was written, but no additional major modifications to TEXT-PAC are anticipated. Dr. Borko's class will use TEXT-PAC also to make searches of a sample tape from the American Institute of Physics.

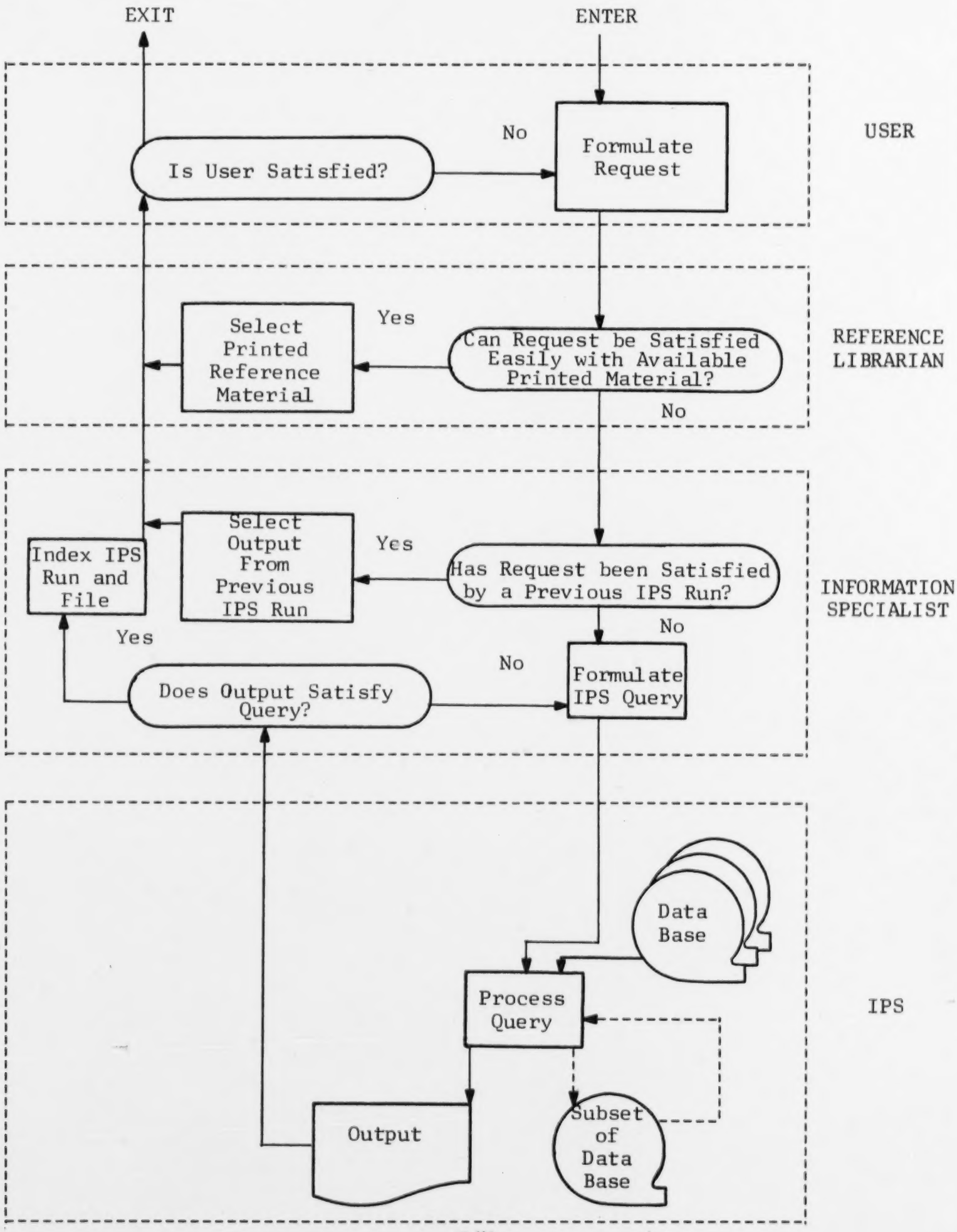
Experience gained with the interim software is expected to provide information valuable to the development of CIS. This experience should affect such areas as operational procedures, profile development, marketing user interaction, and monitoring and evaluation.

USER INTERFACE

The experimental operations with the interim software should provide important information about the user interface problem. A more complete picture can be obtained, however, if consideration is given also to the structure of CIS when it becomes operational. A somewhat simplified view of one aspect of CIS, that of handling user requests, is shown in Figure 7.

While it will be possible for the more sophisticated users to access the IPS directly, it is expected that most users will require the services of an information specialist. Furthermore, since the use of the IPS by

FIGURE 7
OPERATIONAL CIS - HANDLING USER REQUESTS



the information specialist and by the sophisticated user will be similar, the emphasis here will be focused on the former. The user (or his request) may arrive at the desk of an information specialist as the result of a request made of a reference librarian or he may come directly. (Of course, nothing precludes the reference librarian and the information specialist from being the same person.)

As shown in Figure 7, the reference librarians should first determine if a user's request can be satisfied easily using traditional methods. If not, the request is given to the information specialist who check (if possible) an index to determine if a similar request has already been satisfied by a previous IPS run. If so, he may be able to direct the user to the results of this earlier request. Otherwise, the information specialist reformats the request as an IPS query and submits it to the IPS.

The query specifies the data base to be used, and depending upon the nature of the request, a subset of the data base may be produced. The subset can be the results desired by the user or it may be used as input in subsequent IPS runs to refine the results to be given to the user. In any event, the information specialist examines the results on each IPS run to determine if the user's request, as the specialist understands it, has been satisfied. If so, the information specialist may then index the run for future reference and return the results to the user.

Clearly, the procedure can be improved if the information specialist discusses the output of each IPS run with the user. This may or may not be possible depending upon the nature of the request and the timing (i.e., how soon the IPS run can be made after the query if formulated). Also not shown in a feature which would allow a user to browse and choose the output he would like to receive as hard copy. Primarily the diagram is intended to show how CIS can complement the normal activities of a reference librarian and how it might appear to a typical user. This is a first approximation and more work is required in this area.

FMS

FMS is a text formatting system for producing hard copy output from free format text and command input. FMS is faster and cheaper than most text formatting systems and gains its speed by supporting the formatting operations only, i.e., no editing or text scanning operations are included. (At UCLA, the editing and scanning functions can be achieved through the URSA console system.) FMS operates on any IBM 360 using OS/360. It has gained wide acceptance among CCN's user population and has been exported to a number of other schools. The development of FMS was supported in part by the CIS project and is being used by the CIS staff to document their work. Consideration will be given to using FMS to format output produced by the CIS system.

DISCUS

One feature of the software system being designed for CIS is an interactive instructional package from which a user can obtain instruction on how to use the system. The instructional package will be able to do such things as explain the mechanics of the system, assist in writing profiles, help interpret output, and describe the form and content of the various data bases that are available. The package will be written so that new instructions can be added easily.

For this instructional package CIS is considering using DISCUS, a computer-aided instruction language under development by Steve Silver (CCN--UCLA), Joseph Meredith (ILR--UC Berkeley), and Rodney Randall (ILR--UC Berkeley). DISCUS consists of a compiler for the language and an interactive processor for CRT's. Instructional programs can be written in DISCUS, compiled, and then executed interactively. DISCUS is only partially complete but results to date look promising.

NUMERICAL PROCESSING STUDIES

CIS will include in its operational system the ability to process numerical data bases. It is not intended that CIS develop software that duplicates existing mathematical and statistical packages; however, CIS software should provide for extraction and relatively simple processing of numerical data in preparation for later processing by other packages. As an initial effort into this area, the Library has assigned a research assistant to work with the CCN staff in studying problems in numerical processing. The 1970 Census was chosen as a representative (and likely) numerical data base to be processed by CIS.

The research assistant is surveying available statistical packages, determining how to use each, and evaluating them (in terms of ease of use, capabilities, cost, output, etc.). A study will then be conducted to determine how the packages might interface with the CIS software.

CIS staff members have been monitoring the material on the 1970 census that has been released by the Census Bureau, and have attended seminars on census tape processing. The Library has subscribed to a service being offered by DUALabs through the Council of Research Libraries. One feature of this service is a set of programs for processing the census tapes. If appropriate, CIS will install these programs as interim software.

IV. EXPERIMENTAL SDI SERVICE

Beginning January 1, 1971, UCLA will become a licensed center for the Chemical Abstracts Service and will begin experimental operation of an SDI service using the CA Condensates file. The experimental period will be for six to twelve months and initially will serve only users from the Chemistry Department.

CIS has focused its attention on six areas of activity in preparing for the service: software, operating procedures, promotion, user interface, monitoring and evaluation, and administrative matters.

SOFTWARE

The software system to be used is IBM's TEXT-PAC acquired by CIS last summer. A program to convert the CA Condensates file (old format) to the TEXT-PAC format was supplied through the generosity of the University of Pittsburgh. Experience was gained with TEXT-PAC using a sample CA Condensates file while a conversion program for the new Standard Distribution Format (SDF) was written. The conversion program is now complete and the SDI service will be offered using the SDF format.

OPERATING PROCEDURES

Operating procedures have been developed (see Appendix A) which detail the steps to be taken from the time a CA Condensates tape is received until the output is delivered to the users or information specialist for review. Besides outlining the day-by-day activities, the procedures will also include the maintainance (but not the preparation) of profiles, backup for all files, and preparation of files for further retrospective searches.

PROMOTION

Promotional activities have centered around informing the Chemistry Department faculty, staff, and students of the SDI service. A letter describing the service (see Appendix B) was sent from Peter Watson, Coordinator of CIS for the Library, and George Keller, Chemistry Librarian, to Dr. Robert Scott, Chairman of the Chemistry Department. This letter was distributed at a departmental meeting and requested that interested persons contact Mr. Keller at the Chemistry Library. A meeting of

prospective users is planned at which time the service will be explained more fully. Those planning to use the service will then be contacted individually by the CIS staff.

USER INTERFACE

Activities in the user interface area have been concerned primarily with procedures for determing user interest areas, developing profiles, and reviewing and evaluating output with users. The Chemistry Department is contributing a graduate assistant to work with the CIS staff as a profile analyst. A Statement of Interest (see Appendix C) to be completed by the user will assist the analyst in determining the user's areas of interest. Instructions for constructing TEXT-PAC profiles have been prepared and are recorded in Appendix D. Initially all output will be reviewed with the user to determine whether revisions to his profile are necessary.

MONITORING AND EVALUATION

The primary purpose of the experimental operation is to obtain feedback. To facilitate monitoring and evaluation of the CA Condensates service, a set of questionnaires will be distributed, user profile records will be maintained, and a statistical report will be generated to reflect the quality and quantity of the output provided by the TEXT-PAC programs.

There will be two basic questionnaire types. First, individuals will be questioned by personal interview upon expressing an interest in using the system. This interview will last approximately fifteen minutes and the resulting user responses will be compiled to indicate the general make-up of the user population, their respective backgrounds, and their reasons for using the service (see Appendix E). A second class of questionnaires will accompany the user's computer output (his "alerts") for the first five oomputer searches and the user's responses will be used as feedback to evaluate the service provided (see Appendix F). Development of the questionnaires was guided by questionnaires used by the Chemical Information Center at the University of Pittsburgh.

Data gathered from the profiles, computer runs, and user feedback questionnaires will be maintained in the form of User Profile Records (see Appendix G). These records will record search results, user satisfaction, and profile changes.

Finally, the data gathered from the set of user profile records will be combined with the data from the computer search runs (e.g., time required, number of abstracts, etc.) to generate a CA Condensates Search Summary (see Appendix H) which will provide service performance statistics.

ADMINISTRATION

Administrative matters include establishing schedules, determining personnel requirements, and planning when and where to expand and extend service. An overall administrative function is to ensure that the experimental operation follows the schedule established for it and fulfills its goals. There will be no charges to the users during the experimental period, but a statement showing what the charges would have been will be included with each user's output. The particular charging strategy to be used is still under study.

IMPLEMENTATION SCHEDULE

The schedule for the CA Condensates service is shown below:

January 4, 1971:

a. Software modifications complete.
b. CA Condensates tapes begin arriving.
c. Begin shake-down of operational procedures.
d. Establish meeting data for prospective users.
e. Review monitoring and evaluation plans.
f. Begin training of Chemistry Department's graduate assistant.

January 18, 1971:

a. Begin writing and testing profiles for users.
b. Continue shake-down.
c. Test monitoring techniques.

February 2, 1971:

a. Begin production runs.
b. Begin monitoring and evaluation.

APPENDIX A

CHEMICAL ABSTRACTS OPERATING PROCEDURES

Note:

(1) The names of data sets used for JCL, profiles, etc. for even issues differ from the names of data sets used for odd issues only in the third index level. Therefore data set names are given with the entire data set name for even issues followed by the third index level for odd issues in parenthesis. For example, "CCB006.CIS.PRFLEVNX(PRFLODDX)" indicates data set CCB006.CIS.PRFLEVNX for even issues and CCB006. CIS.PRFLODDX for odd issues.

(2) In data set names, volume serial numbers, etc. lower case letters should be replaced with appropriate values. For example, "CCB006.CIS.CASDF.VvvIii" indicates that "CCB006.CIS.CASDF.V74I01" will be used for Volume 74, Issue 1.

Chemical Abstracts Operating Procedures

DAY 1

1. The profile analyst should finish writing new profiles and revising profiles and turn them in at BH 3531.

2. Set up address labels for new profiles.

3. Check for available tapes for this week's run (need 1 every week and 1 more every fourth week) and if more are needed, submit job (CCB006LL) to relabel. The volume serial number (CISnnn) should be assigned by the Tape Librarian. Be sure to update the volume label in card 006 of the deck.

Chemical Abstracts Operating Procedures

DAY 2

1. Assign profile numbers to any new profiles. Profile numbers are of the form aafuup where:

   aa is 'OD', 'EV', or 'BT' to indicate whether the profile is to be run against odd issues only, even issues only, or both odd and even issues.

   f is a code for the field of chemistry:

   B - Biochemistry
   E - Applied Chemistry and Chemical Engineering
   M - Macromolecular Chemistry
   O - Organic Chemistry
   P - Physical and Analytical Chemistry

   uu is a number assigned to the user and is the same for all his profiles.

   p is used to sequentially number the profiles of each user, starting at 0.

   After determining the left most three digits of the number, (aaf), look at the Keypunch data set CCB006.CIS.PRFLLOG which contains for each profile (in ascending order by profile number) the profile number, the users name and the profile text sequential number after his last profile. If this is the users first profile, assign it the number aafuup where uu is the next user number within the "aaf" group and p is zero. Enter the new profiles in the data set as the profile numbers are determined.

2. Using the URSA KEYPUNCH service, enter new profiles and revisions to CCB006.CIS.PRFLEVN (PRFLODD) for profiles to be run against even (odd) issues only and to CCB006.CIS.PRFLBTH for profiles to be run against both even and odd issues.

DAY 2 (Continued)

The profiles must be in sequential order within each data set.

Be sure to update the effective volume and issue in the header card of revised profiles.

3. Submit (through URSA) the profile diagnostic job CCB006.CIS.PRFLEVNX(PRFLODDX).

4. Correct any errors found in the profiles and check the listing for omissions and non-syntactic errors. Resubmit CCB006.CIS.PRFLEVNX(PRFLODDX) until there are no errors. Make a xerox of each new or revised profile (underline changes) and file in user file.

5. Scratch (using URSA's UTILITY service) CCB006.CIS.BKUPEVN (BKUPODD) and CCB006.CIS.BKUPBTH. Using URSA's MOVE service make a new backup of CCB006.CIS.PRFLEVN(PRFLODD) called CCB006.CIS.BKUPEVN(BKUPODD) on STAFF1 and a new backup of CCB006.CIS.PRFLBTH called CCB006.CIS.BKUPBTH on STAFF1.

Chemical Abstracts Operating Procedures

DAY 3

1. Receive tape (mail room in Boelter Hall will call us or deliver it in person).

    a. Log in on CA Tape Log Sheet (attached).

    b. Paste on external label, for example:

    Vol ID 740101
    Tracks: 7_____ 9 XX
    Volume Has: Standard Label XX No Label___
    Chge # CCB006 Ext _____
    User Name: CIS, L. MIROFF or A. DE BOER

2. Update the copy deck (CCB006LC)

    a. Card 3 - the volume serial name of the CA tape (vviirr).

    b. Card 6 - the file number and/or volume serial name of the copy tape (see tape log).

    c. Card 5 - the issue and/or volume in the data set name (CCB006.CIS.CASDF.VvvIii).

Submit the deck. After it is run, retrieve the original tape from the return rack and store in BH 3531-B

Chemical Abstracts Operating Procedures

DAY 4

1.

a. Update CCB006.CIS.CR1EVNX(CVR1ODDX)

line 120 - the volume and issue in the parm field (PARM='CABvvii00001')

line 200 - the data set name of the SDF copy (CCB006.CIS.CASDF.VvvIii)

line 220 - the file number and/or the volume serial of the copy tape

line 240 - the data set name of the converted tape (CHEMAB.CNVRTEXT.VvvIii).

Submit CCB006.CIS.CVR1EVNX(CVR1ODDX)--routed to URSA; when it completes, have it printed and (if successful) go on to (b). The figure called "RECORDS WRITTEN" should be noted on the CA Tape Log as number of citations.

If the figure called "ERROR RECORDS WRITTEN" is 3, all citations were converted successfully.

b. Update CCB006.CIS.CVR2EVNX(CVR2ODDX)

line 200 - the data set name of the converted tape (CHEMAB.CNVRTEXT.VvvIii).

line 240 - the data set name of the output of TRC260 (CHEMAB.DATABASE.VvvIii).

Submit CCB006.CIS.CVR2EVNX(CVR2ODDX)--routed to URSA; when it completes, have it printed and (if successful) go on to (c).

DAY 4 (Continued)

c. Update CCB006.CIS.CVR3EVNX(CVR3ODDX)

line 200 - the data set name of the output from TRC260 (CHEMAB.DATABASE.VvvIii).

lines 240 and 380 - the data set name of the search text (CHEMAB.SRCHTEXT.VvvIii).

line 420 - the data set name of the converted tape (CHEMAB.CNVRTEXT.VvvIii).

lines 460 and 480 - the data set name of the master tape (CCB006.CIS.CATPM.VvvIii), the file number and/or the volume serial (CISnnn).

Submit CCB006.CIS.CVR3EVNX(CVR3ODDX)--routed to URSA; when it completes, have it printed and (if successful) go on to (2).

2. Update CCB006.CIS.SRCHEVNX(SRCHODDX)

lines 200 and 220 - the data set name of the master tape (CCB006.CIS.CATPM.VvvIii), the file number and/or the volume serial (CISnnn).

Submit CCB006.CIS.SRCHEVNX(SRCHODDX).

3.

a. Enter CCB006.CIS.CASDF.VvvIii and CCB006.CIS.CATPM.VvvIii on the Tape Librarian's log.

b. Update the issue and/or volume in the following data set names on the Tape Librarian's log:

	Volume Serial	
	Even Issues	Odd Issues
CHEMAB.CNVRTEXT.VvvIii	CIS005	CIS002
CHEMAB.DATABASE.VvvIii	CIS006	CIS003
CHEMAB.SRCHTEXT.VvvIii	CIS007	CIS004

Chemical Abstracts Operating Procedures

DAY 5

1. Profile analyst screens output from search and marks alerts to be eliminated. He may wish to refer to the user's Statements of Interest (particularly items 10, 18, and 19).

2. Punch CCB006.CIS.TRIGEVN(TRIGODD); separate the trigger cards for the alerts to be eliminated (as indicated by the profile analyst above).

   Place the deck of good trigger cards behind card 008 of CCB006LE(CCB006LO) and submit it to print the final output.

3. Record the number of bad trigger cards for each profile on the User Profile Record under "Analyst rejects". Record the total number of alerts produced under "# alerts".

Chemical Abstracts Operating Procedures

DAY 6 (DAY 1, Tape N-1)

1. Check that the final output contains the correct citations according to the marked, preliminary output.

2. Burst the output (EVB010, EVB011, . . . go to the same user); stick on the labels and deliver. Chemistry Department output can be delivered to Chemistry (Young Hall) 3034.

APPENDIX B

ANNOUNCEMENT OF EXPERIMENTAL SERVICE

TO: Professor R. L. Scott, Chairman, Department of Chemistry

FROM: George Keller, Librarian, Chemistry Library
Peter Watson, Institute of Library Research

SUBJECT: An experimental information service from Chemical Abstracts

DATE: December 4, 1970

The Institute of Library Research on this campus has been coordinating a joint project (under NSF funding) between the Institute, the Campus Computing Network and the University Library to design a Center for Information Services (CIS) which would enable the Library to acquire and provide service from all types of computer processible data bases. This is a 5-7 year program of research and development which is now entering a critical phase, namely the transition from purely internal activity to the beginning of service to our immediate user community--the UCLA faculty and staff.

One of the types of services CIS plans to provide is current awareness. Current awareness services help professional people keep abreast of current research and developments by providing the latest bibliographic citations in their specific field statements of interest, called 'profiles'. The CA Condensates tapes of the Chemical Abstracts Service, which contain searchable information from the corresponding printed issues of Chemical Abstracts, have been selected as the primary vehicle for testing our ability to mount a current awareness service--our subscription to CA Condensates will be effective 1 January 1971.

Accordingly, we are inviting members of the Chemistry Department faculty to participate as users in this experimental current awareness service, before broadening its scope to the campus as a whole later in 1971. Every participating faculty member will be able to submit statements of interest corresponding to his own subject field, and/or the topic of a particular course, and/or the research area of a specific graduate project. The Chemistry Department has committed one of its Teaching Assistants to work with the CIS staff as a search

analyst coding the profiles for input to the computer and interfacing with participating faculty. In the initial period, participants will be asked to evaluate the service for its relevance, timeliness, scope, and general fulfillment of their information needs.

Normally, there is a charge for this service; however, during the initial experimental period (6-12 months) there will be no charge to Chemistry Department users. It will probably be necessary to assess a modest charge after this period.

Would you please ask interested faculty members to signify their willingness to participate by notifying the Chemistry Library before 15 January? They will then be contacted and instructed further on the procedure for submitting interest statements, and we shall be aiming to start the service on or about 1 February.

Thank you for your assistance in this venture; we look forward to working with members of the Chemistry Department.

GK,PW:j

cc: Mr. Bruce Briggs
Mr. R. L. Carmichael
Mr. W. B. Kehl
Mr. R. Vosper

APPENDIX C

USER STATEMENT OF INTEREST

UNIVERSITY OF CALIFORNIA
LOS ANGELES

CENTER FOR INFORMATION SERVICES

1. Submitted by _____ Date _____

2. Is this your first Chemical Abstracts Profile? Yes___No___

3. Position _____ Department _____

4. Campus Address _____ Office Hours _____

5. Telephone _____

6. Individual(s) who will actually use the service (if different from above)

_____

7. Position _____ Telephone _____

8. Profile analyst (leave blank) _____

9. Profile Title (short descriptive phrase) _____

_____

USER STATEMENT OF INTEREST (Continued)

10. Detailed statement of interest area: please describe, in your own words, the subject you are interested in and want references on. Be specific and define terms which have special meaning in your request. If certain facets of the subject are not of interest to you, explicity exclude them.

_____

_____

_____

_____

_____

_____

_____

11. For the important terms (words and word phrases) in your statement of interest, list synonyms or any special uses you wish to exclude. Be sure to include both scientific and common terms. Your careful completion of this table will help us to refine your profile and give you the references which are valuable to you.

TERMS	SYNONYMS OR CLOSELY RELATED TERMS	EXCLUDED USES

USER STATEMENT OF INTEREST (Continued)

12. Information purpose (general current awareness, a special project, a class, etc.)

_____

13. Do you want your profile to be run against

_____ Odd issues (biochemistry and organic chemistry).

_____ Even issues (macromolecular chemistry, applied chemistry, chemical engineering, physical and analytical chemistry).

_____ Both odd and even issues.

Please note that the odd and even issues of the tape correspond in coverage to the odd and even issues of the printed Chemical Abstracts. Therefore, you may refer to an issue of Chemical Abstracts (available in the Chemistry Library or the Engineering and Math Science Library) for more information on Chemical Abstracts coverage.

14. List any particular authors whose writings always interest you (specify first and middle names or initials if possible).

_____ _____ _____

_____ _____ _____

_____ _____ _____

15. List any journals or other periodicals which you regularly read and whose contents you do not need to be informed of.

_____ _____

_____ _____

_____ _____

USER STATEMENT OF INTEREST (Continued)

16. Check any of the following types of references which you do not wish to receive:

_____ Patents
_____ Dissertations
_____ Conferences
_____ Technical Reports
_____ Books
_____ Periodical Articles

17. Do you wish to receive references to documents written in

_____ English
_____ Any Language
_____ Other Languages (specify)

_____
_____
_____

18. Do you wish to receive

_____ Nearly all possible relevant references
_____ Fewer, very probably relevant references.

18. What is the maximum number of references you would wish to receive in any one weekly run? _____

APPENDIX D

WRITING TEXT-PAC PROFILES

UNIVERSITY OF CALIFORNIA
LOS ANGELES

CENTER FOR INFORMATION SERVICES

Introduction

Text-Pac is designed to search the words of a bibliographic citation or of an entire document to determine if they satisfy logical expressions which are defined in the profile and are called concepts. The number of concepts which must be matched for a document to be selected is specified in the profile header card and is called the match criterion.

A concept, which is identified by an alphanumeric symbol whose first three letters are CON (e.g., CON1, CON2), is a logical expression involving either words and word stems or logic symbols which identify logical sub-expressions called logic levels. The following is an example of a simple concept involving only words:

```
CON1    SEISMOLOGY OR SEISMIC
```

A logic level, which is given an alphanumeric name (called a logic symbol) with an alphabetic first character (e.g., A1, A2) also consists of a logical expression involving either words and word stems or logic symbols. In the following example A1 and A2 are logic symbols used in CON1 which is a concept:

```
A1      SEISMIC ADJ WAVES
A2      SOUND ADJ WAVES
CON1.   A1 OR A2
```

The form of the logical expressions which make up the logic levels and concepts is described below.

The basic elements of the logical expressions are single words or word stems:

A. Any word consisting of up to 38 characters may appear within a profile. However, only the first 20 characters are used for searching.

B. Word stems instead of complete words may be used for searching by indicating either of two types of truncation (masking):

1. Selective truncation operates on the stem of a word and a specified number of characters past the stem. Selective truncation is indicated by one to six dollar signs with the number of allowable characters past the stem corresponding to the number of dollar signs. For example, OXID$$ would match OXIDE or OXIDES, but not OXIDATION.

2. Unconditional truncation is indicated by $* and indicates all words with the same stem. For example, POLYMER$* matches POLYMERS, POLYMERIZED, POLYMERIZATION.

The connectors which serve to express logical or positional relationships between the words and word stems are OR, AND, ADJ, and WITH.

A. The logical connector OR can be used to indicate that any of the connected profile words or logical symbols will satisfy the logical expression. Synonyms or equivalent expressions are usually connected by OR's. For example,

A1 ALCOHOLI$$ OR DIPSOMANIA$ OR DIPSORRHEXIA

B. The logical connector AND can be used to connect words (or logical symbols) which must all be matched by the citation for an alert to occur. For example,

CON1 POLLUTION AND PESTICIDES

specifies that the citation must include the word POLLUTION and the word PESTICIDES.

AND may also be used to specify that two or more sub-expressions be jointly satisfied. The example below requires that one of the alternative words for pollution and also one of the alternative words for pesticides be present in a selected citation.

A1 POLLUTION OR SMOG OR CONTAMINATION
A2 PESTICIDES OR INSECTICIDES
CON1 A1 AND A2

C. Adjacent words: a phrase of two or more words or word stems may be specified by connecting the words or word stems with the operation ADJ. The words must appear in the citation in the identical sequence to be matched. For example,

```
CON1  INFORMATION ADJ RETRIEVAL
```

would be matched by INFORMATION RETRIEVAL but not RETRIEVAL INFORMATION.

ADJ can also be used to connect two or more logical symbols which denote logical sub-expressions, if the logical sub-expressions consist only of words jointly by OR logic. For example,

```
A1    INFORMATION OR DOCUMENT
A2    RETRIEV$$$ OR SEARCH$*
A3    SYSTEM$ OR CENTER$ OR SERVICE$
CON1  A1 ADJ A2 ADJ A3
```

D. Co-occuring words: the logical connector WITH can be used to cause a match if the connected profile words are found in the same sentence of the document. For example,

```
CON1  OXIDATION WITH METALS
```

Like ADJ, WITH can also be used to connect two or more logical symbols which denote logical sub-expressions. However logical sub-expressions used in this way may only consist of words joined by OR logic. For example,

```
A1    INFORMATION OR DOCUMENT$
A2    RETRIEVAL OR SEARCH$*
CON1  A1 WITH A2
```

Generally only one type of logical connector is permitted within a concept or logic level. However, OR groups of words or word stems may occur within ADJ or WITH logic. Thus the following concept would be matched by STATISTICAL METHODS, MATHEMATICAL ANALYSIS, PROBABILITY THEORY, etc.

```
CON1  STATISTIC$$ OR MATHEMATIC$$ OR PROBABILITY
      ADJ METHOD$ OR TEST$ OR ANALYSIS OR THEORY
```

Replacing ADJ with WITH would also yield a correct expression.

The logical expression of a concept may be preceeded by one of the logical modifiers NOT and ABS.

A. The NOT modifier denotes profile words or logic levels which should not cause selection. A concept containing NOT overrides all other concepts--except those containing ABS. For example, if we need information on the manufacturing or characteristics of synthetic foam, but want to exclude

documents dealing with the marketing of synthetic foam, the following profile could be used:

```
A1    SYNTHETIC OR PLASTIC ADJ FOAM$
A2    PROPERT$$$ OR CHARACTERISTIC$ OR MANUFACTUR$$$ OR
      PRODUC$$$$
CON1  A1 AND A2
CON2  NOT MARKET$$$ OR SALE$ OR BUY$$$ OR CONSUM$*
```

B. The logical modifier, ABS, indicates that a document which satisfies the logical expression which follows will always be selected regardless of any other logic (such as a match criteria or of more than 1 or another concept containing NOT). For example, if the following concept is included in a profile any citation containing "ALPHA EMMISSION" will be selected,

```
CON1  ABS ALPHA ADJ EMISSION.
```

Print Controls

A. The various parts of Text-Pac citations or documents are assigned identification numbers, called print controls. For the Chemical Abstracts Condensates tapes, the following print controls are used:

00	Title (original - foreign)
09	CAS citation with publication class
10	Coden
20	Authors
21	Location of work
22	Patent assignees
30	Journal citation or Publication information or Patent information
41	Copy abstract citation
42	Parent citation
43	Book review citation
60	Chemical Abstracts Section
61	Keyword phrases

B. Text-Pac allows searching to be done either on all print control fields (as in previous examples), on only specified print control fields, or on all print control fields except those specified.

1. Control logic is used to specify a match on a word or phrase only if it appears in the indicated control fields. For example, to search for documents with R. Sachsse, as author, we could use the following concept:

```
CON1  SACHSSE CONTROL20 ADJ R
```

2. Negative control logic specifies a match for any control field except the one indicated. For example, in order to search for literature on therapy but not match on the names of institutions such as Smith Therapy Center which might appear in print control field 21, we could exclude searching on this field:

```
CON;  THERAPY NOT-CONTROL21
```

Up to 7 print controls may be specified on a word. The print controls must be in ascending order:

```
A1    THERAPY NOT-CONTROL21,30
CON1  SACHSSE CONTROL20,22 ADJ R
```

Print controls may be masked on the second and third digits. Note that even though all the Chemical Abstracts Condensates print controls are only two digits, if the second digit is masked the third one must also be masked. For example,

```
A1    STATISTICAL NOT-CONTROL4$$
```

is equivalent to

```
A1    STATISTICAL NOT-CONTROL41,42,43
```

Note that CONTROL and NOT-CONTROL can only be used with words not with logic symbols.

Rules and Limitations

The most important rules and limitations for Text-Pac profiles are summarized below:

1. Only one type of logical connector (OR, AND, ADJ, and WITH) is permitted within each logic level or concept. The only exception is that OR groups (of words only) can be used within ADJ and WITH logic.

2. Logic symobls (such as A1 and CON1) and words cannot be mixed within a logical expression.

3. No more than three levels of back-referencing are allowed; if a logical symbol in a logical expression defines another conjunction of logical symbols, this is considered two levels of back-reference. The following example contains two levels:

```
A1     ENERGY ADJ LEVELS
A2     QUANTUM ADJ LEVELS
A3     A1 OR A2
A4     ELECTROCONIC ADJ SHELL
CON1   A3 AND A4
```

Back-referencing to concepts is not allowed.

4. Control and negative control logic can only modify words, not logic symbols. Up to 7 print controls can be specified (in ascending order) on each word.

5. When using ADJ or WITH logical connectors you may only use CONTROL or NOT-CONTROL with the first profile word to the left of the first ADJ or WITH. For example,

```
A1     JONES CONTROL20 ADJ J ADJ P
```

6. When using ADJ or WITH as connectors between logic symbols, the expressions referenced can only contain query words combined with OR logic.

7. ABS and NOT modifiers can appear only in concepts, not in logic levels and must be the first word of the logical expression.

8. Hyphenated words are treated as two words.

9. Periods embedded in words must be specified. But trailing periods should not be specified. For example, 'I.B.M.' should be coded 'I.B.M'.

10. No more that 15 logic symbols may be connected in any one logic level. Also no logic symbol can be referenced more than 15 times in the profile.

11. When ANDing words, no more than 15 words may be connected.

APPENDIX E

INITIAL USER INTERVIEW QUESTIONNAIRE

The following questions will be asked of users at an initial interview. The questions will be asked by a member of the CIS Staff and should require approximately fifteen minutes.

The users' responses will be used as a background data base from which the staff will draw for better understanding of the user population and to effect service improvements.

UNIVERSITY OF CALIFORNIA

LOS ANGELES

CENTER FOR INFORMATION SERVICES

INITIAL USER QUESTIONNAIRE

(1) Field of Chemistry

1. Macromolecular Chemistry
2. Applied Chemistry and Chemical Engineering
3. Physical and Analytical Chemistry
4. Biochemistry
5. Organic Chemistry

(2) Have you had any previous experience with computer-based current-awareness or information-retrieval systems? If so, what were they?

(3) What do you expect to gain from the system?

_____save time

_____broader coverage

_____both

(4) What do you consider to be the longest time-lag that you are willing to tolerate between the time that you receive your alerts and the time that the publications are available?

(5) Approximately how many hours per week do you spend keeping up with developments in the literature of your field? Of this total, how much is searching time and how much is reading time?

Total_____

Searching_____

Reading_____

(6) How much time every two weeks do you feel that you can expend in using our system? That is, modifying profiles, communication, answering questions, etc.

(7) How much time every two weeks can you expend in reviewing your computer output?

(8) Are you using the system for yourself or on behalf of a group?

(9) How many times per month do you consult a librarian for any sort of assistance in keeping abreast of current publications in your field?

(10) What <u>level</u> of information do you consider most useful in your work?

_____citation

_____abstract

_____source document

(11) To how many scientific journals do you personally subscribe? (include any published subsections of Chem Abstracts).

(12) Do you have any apprehensions about using the system? If so, what are they?

APPENDIX F

FOLLOW-UP QUESTIONNAIRES

Questionnaires 1-5

After the initial interviews have been completed, the user will receive, with each of his first five sets of output (alerts), a set of questions to measure his satisfaction with the service and his recommendations for improvements. This feedback will enable the CIS Staff to alter their procedures if necessary.

There will be five questionnaires, and a subset of the following questions will be asked on all five questionnaires. The numbers in parentheses indicate the questionnaire numbers for the respective questions (e.g., (2,5) indicates the questions will be asked on questionnaires two and five).

Basically, questionnaires one and four and questionnaires two and five are alike.

FOLLOW-UP QUESTIONNAIRES:

(The numbers in parentheses to the left of each question indicate the numbers of the questionnaires on which that question will appear. There are five questionnaires.)

(1-5) [#1] Thus far do you have any problems in using the system?

(1-5) [#2] Of the alerts that you received with this bulletin, how many do you feel are:

relevant_____

unsure_____

irrelevant_____

(1-5) [#3] Have you discovered a useful article that has not been included in your recent bulletins? We are referring to the articles from the same time period as covered by the bulletins.

(1-5) [#4] Do you wish to change your profile?

(1) [#5] Estimate the extent to which you relied on manual searches of CT and CA before becoming a user of CIS.

(1,4) [#6] What do you consider would be the maximum number of citation alerts which you could fruitfully evaluate each computer run?

(1,4) [#7] If you have received scientific information from any source during the past several weeks that had a major impact on your work, please tell us the source of the information.

(2,5) [#8] How much time per computer run do you actually spend using the alerting system, e.g., profile modification, general communication, answering questions, etc.

(2,5) [#9] After deciding on which of the alerts in this bulletin were useful, how long do you estimate you spent:

obtaining the documents_____

reading and evaluating the documents_____

(2,5) [#10] What percentage of the total number of alerts that you receive are you willing to tolerate as being not useful for the interest covered in your profile, i.e., what percentage are you willing to weed out?

(2,5) [#11] Estimate the percentage of the information that you receive on your CA Condensates search that is redundant, i.e., information that you have already seen, either in your manual searches or browsing in the chemical literature.

(2,5) [#12] To what extent is the CIS service helpful to your work?

not useful_____

occasionally useful_____

frequently useful_____

(2,5) [#13] Have you thus far saved significant time and effort by using our computer-based information system?

APPENDIX G

USER PROFILE RECORDS

To facilitate bookkeeping and monitoring of user progress, User Profile Records will be maintained. There will be on record for each profile and it will be used to record the users profile changes, the results of his computer searches, and the success of his alerts in satisfying his profile interests.

FIGURE G-1

USER PROFILE RECORD

User_____

Profile Number_____

Date_____

Profile_____

Profile Changes:

Date_____

Profile_____

Date_____

Profile_____

Issue Number	$issue_1$	$issue_2$	. . .	$issue_n$
Number of Concepts			. . .	
Number of Alerts			. . .	
Relevant			. . .	
Irrelevant			. . .	
Analyst Rejects			. . .	
User Rejects			. . .	

APPENDIX H

CA-CONDENSATES SEARCH SUMMARY

This summary reflects the statistics accumulated from the search runs and the profile analysis. Initially the data will be gathered manually, but eventually a package may be included in the computer run to collect this data automatically.

FIGURE H-1

CA CONDENSATES SEARCH SUMMARY

Issue	issue$_1$	issue$_2$	. . .	issue$_n$	averages
No. abstract (before/after conversion)			. . .		
Time - total			. . .		
Reformat (Conversion)			. . .		
Search			. . .		
No. of profiles			. . .		
No. of concepts			. . .		
No. of terms			. . .		
Range (high/low)			. . .		
No. of alerts			. . .		
Range (high/low)			. . .		
Terms/profile			. . .		
Alerts/profile			. . .		
Alerts/term			. . .		
Search time/term (sec.)			. . .		
Total cost			. . .		
Reformat			. . .		
Search			. . .		
Cost/profile			. . .		